CLAIMS

What is claimed is:

1. A method for loop-free multipath routing in a network of interconnected router nodes, comprising:

computing shortest multipath loop-free route distances between a source and corresponding destination using loop-free invariant conditions; and

exchanging distance values among neighboring routers;

wherein said loop-free invariant conditions prevent a count-to-infinity problem and ensure termination of said computing of loop-free route distances.

2. A method as recited in claim 1, further comprising:

generating a routing graph from said route distances.

3. A method as recited in claim 1, further comprising:

if the distance increases for a route, executing a diffusing computation.

4. A method as recited in claim 1, further comprising:

providing multiple next-hop choices for each destination.

5. A method as recited in claim 1, wherein nodes exchange messages containing distance information to maintain a routing table at each node.

6. A method as recited in claim 1, wherein ordering of messages from rapidly changing sources is supported for overlapping receiver groups and for anonymous hosts.

7. A method as recited in claim 1, further comprising:

distributing ordering among a plurality of nodes across a logical tree.

8. A method as recited in claim 7, further comprising:

using aggregation of ordering primitives to minimize control traffic among nodes.

9. A method as recited in claim 7, further comprising:

using address extensions assigned to hosts for self-routing of messages and dynamic distribution of processing load for said ordering.

10. A method as recited in claim 9, further comprising:

using said address extensions, supporting total ordering of messages for anonymous and overlapping receiver groups in shared trees.

11. A method for loop-free multipath routing in a network of interconnected router nodes, comprising:

computing shortest multipath loop-free route distances between a source and corresponding destination according to loop-free invariant (LFI) conditions that prevent

a count-to-infinity problem and ensure termination of said computing of said loop-free route distances;

exchanging distance values among neighboring routers; and

if the distance increases for a route, executing a diffusing computation.

12. A method as recited in claim 11, further comprising:

generating a routing graph from said route distances

13. A method as recited in claim 11, wherein nodes exchange messages containing distance information to maintain a routing table at each node.

14. A method as recited in claim 11, wherein ordering of messages from rapidly changing sources is supported for overlapping receiver groups and for anonymous hosts.

15. A method as recited in claim 11, further comprising:

distributing ordering among a plurality of nodes across a logical tree.

16. A method as recited in claim 15, further comprising:

using aggregation of ordering primitives to minimize control traffic among nodes.

17. A method as recited in claim 15, further comprising:

using address extensions assigned to hosts for self-routing of messages and dynamic distribution of processing load for said ordering.

18. A method as recited in claim 17, further comprising:

using said address extensions, supporting total ordering of messages for anonymous and overlapping receiver groups in shared trees.

19. A method of determining loop-free multipath routes within a network of interconnected router nodes executing a routing protocol, comprising:

compute link distance between a source and destination;

exchanging distance and status information between said nodes;

executing a diffusing computation if the distance of a link to a destination increases;

maintaining a set of routing tables containing information about distance, neighbors, and links within said network based on information exchanged with other nodes; and

selecting a loop-free route according to a set of loop-free invariant (LFI) conditions.

20. A method as recited in claim 19, further comprising:

exchanging said distance and status information using messages containing at

least one entry of the form *[type,* j , d *]*;

wherein d is the distance of the node sending the message to destination j and *type* is the message type; and

wherein *type* is selected from a group of message types consisting essentially of QUERY, UPDATE, and REPLY.

21. A method as recited in claim 19:

wherein said diffusing computation is executed by sending query messages to neighbors with the best distance through the subset of neighboring nodes S_j^i.

22. A method as recited in claim 19:

wherein said nodes remain in a PASSIVE state and enter an ACTIVE state to engage in a diffusing computation; and

wherein if the increase in distance is the result of a query from a successor, said neighbor is added to the list of neighbors waiting for replies QS_j^i to provide a reply when the node transitions to a PASSIVE state.

23. A method as recited in claim 19, wherein the information within said routing tables comprises:

distances to neighboring nodes;

successor sets for each destination, or equivalent;

feasible distance for each destination, or equivalent;

reported distance for each destination, or equivalent;

shortest possible distance through the successor set for each destination, or equivalent;

a set of neighbors engaged in a diffusing computation; and

cost of adjacent links.

24. A method as recited in claim 19, wherein said routing tables comprise a main table, a neighbor table, and a link table.

25. A method as recited in claim 24:

wherein said main table comprises storage for the link distance D_j^i to the destination.

26. A method as recited in claim 24:

wherein said main table comprises storage for successor set S_j^i, feasible distance FD_j^i, reported distance RD_j^i, and shortest distance through successor set SD_j^i, and the set of neighbors involved in a diffusing computation $QS_j^i \subseteq S_j^i$.

27. A method as recited in claim 24:

wherein said neighbor table for each neighbor which contains the distance of neighboring nodes to the destination D_{jk}^i.

28. A method as recited in claim 24:

wherein said link table stores the cost of adjacent links to each neighbor l_k^i.

29. A method as recited in claim 28:

wherein if a link is down its cost is considered to be infinity and the distance to unreachable nodes is also considered to be infinity.

30. A method as recited in claim 19:

wherein said LFI conditions require that for each destination j, a node i can choose a successor whose distance to j, as known to i, is less than the distance of node i to j that is known to its neighbors.

31. A method as recited in claim 30, wherein said LFI conditions comprise:

$$FD_j^i(t) \le D_{ji}^k(t) \text{ while } k \in N^i;$$

where $FD_j^i(t)$ is the feasible distance from node i to node j at time t, $D_{ji}^k(t)$ is the distance of node j to node i as reported by neighbor k which is within the set of neighbors N^i for node i;

where $S_j^i(t) = \{k \mid D_{jk}^i(t) < FD_j^i(t)\}$; and

where $S_j^i(t)$ is a subset of N^i that node i forwards packets to node j, $D_{jk}^i(t)$ is the distance of node k to node j as reported by node i.

32. A method as recited in claim 19, further comprising executing a distributed Bellman-Ford (DBF) algorithm to compute said link distance.

33. A method as recited in claim 19, further comprising generating a routing graph for said nodes within said network;

34. A method of determining loop-free multipath routes within a network of interconnected router nodes executing a routing protocol, comprising:

executing a distributed Bellman-Ford (DBF) algorithm to compute link distance;

exchanging distance and status information between said nodes;

executing a diffusing computation if the distance of a link to a destination increases;

maintaining a set of routing tables containing information about distance, neighbors, and links within said network based on information exchanged with other nodes; and

selecting a loop-free route according to a set of loop-free invariant (LFI) conditions.

35. A method as recited in claim 34, further comprising generating a routing graph SG_j for said nodes within said network;

36. A method as recited in claim 34, further comprising:

exchanging said distance and status information using messages containing at least one entry of the form *[type, j , d]*;

wherein d is the distance of the node sending the message to destination j and *type* is the message type; and

wherein *type* is selected from a group of message types consisting essentially of QUERY, UPDATE, and REPLY.

37. A method as recited in claim 34:

wherein said diffusing computation is executed by sending query messages to neighbors with the best distance through the subset of neighboring nodes S_j^i.

38. A method as recited in claim 34:

wherein said nodes remain in a PASSIVE state and enter an ACTIVE state to engage in a diffusing computation; and

wherein if the increase in distance is the result of a query from a successor, said neighbor is added to the list of neighbors waiting for replies QS_j^i to provide a reply when the node transitions to a PASSIVE state.

39. A method as recited in claim 34, wherein the information within said routing tables comprises:

distances to neighboring nodes;

successor sets for each destination, or equivalent;

feasible distance for each destination, or equivalent;

reported distance for each destination, or equivalent;

shortest possible distance through the successor set for each destination, or equivalent;

a set of neighbors engaged in a diffusing computation; and

cost of adjacent links.

40. A method as recited in claim 34, wherein said routing tables comprise a main table, a neighbor table, and a link table.

41. A method as recited in claim 40:

wherein said main table comprises storage for the link distance D_j^i to the destination.

42. A method as recited in claim 40:

wherein said main table comprises storage for successor set S_j^i, feasible distance FD_j^i, reported distance RD_j^i, and shortest distance through successor set SD_j^i, and the set of neighbors involved in a diffusing computation $QS_j^i \subseteq S_j^i$.

43. A method as recited in claim 40:

wherein said neighbor table for each neighbor which contains the distance of

neighboring nodes to the destination D^i_{jk}.

44. A method as recited in claim 40:

wherein said link table stores the cost of adjacent links to each neighbor l^i_k.

45. A method as recited in claim 44:

wherein if a link is down its cost is considered to be infinity and the distance to unreachable nodes is also considered to be infinity.

46. A method as recited in claim 34:

wherein said LFI conditions require that for each destination j, a node i can choose a successor whose distance to j, as known to i, is less than the distance of node i to j that is known to its neighbors.

47. A method as recited in claim 46, wherein said LFI conditions comprise:

$$FD^i_j(t) \leq D^k_{ji}(t) \text{ while } k \in N^i;$$

where $FD^i_j(t)$ is the feasible distance from node i to node j at time t, $D^k_{ji}(t)$ is the distance of node j to node i as reported by neighbor k which is within the set of neighbors N^i for node i;

where $S^i_j(t) = \{k \mid D^i_{jk}(t) < FD^i_j(t)\}$; and

where $S^i_j(t)$ is a subset of N^i that node i forwards packets to node j, $D^i_{jk}(t)$ is

the distance of node k to node j as reported by node i.

48. A method of determining loop-free multipath routes within a network of interconnected router nodes executing a routing protocol, comprising:

compute link distance between a source and destination;

exchanging distance and status information between said nodes;

executing a diffusing computation if the distance of a link to a destination increases;

maintaining a set of routing tables containing information about distance, neighbors, and links within said network based on information exchanged with other nodes; and

selecting a loop-free route according to a set of loop-free invariant (LFI) conditions;

wherein said LFI conditions comprise:

$$FD_j^i(t) \leq D_{ji}^k(t) \text{ while } k \in N^i;$$

where $FD_j^i(t)$ is the feasible distance from node i to node j at time t, $D_{ji}^k(t)$ is the distance of node j to node i as reported by neighbor k which is within the set of neighbors N^i for node i;

where $S_j^i(t) = \{k \mid D_{jk}^i(t) < FD_j^i(t)\}$; and

where $S_j^i(t)$ is a subset of N^i that node i forwards packets to node j, $D_{jk}^i(t)$ is the distance of node k to node j as reported by node i.